

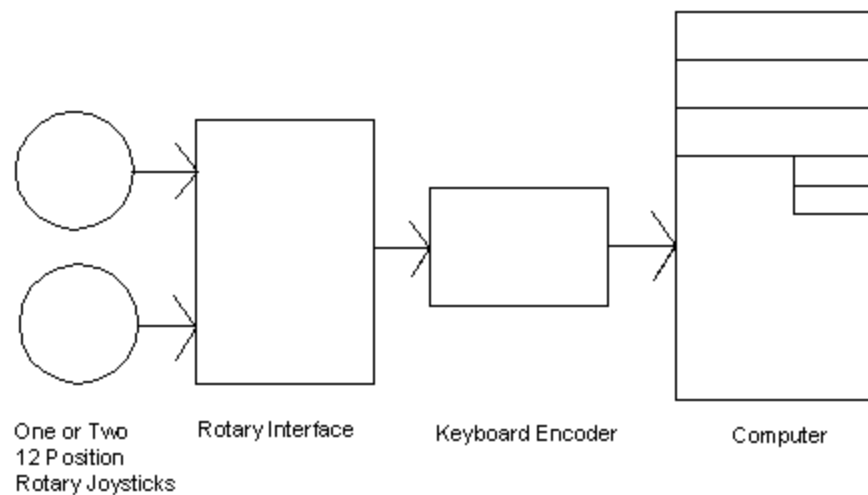
12 Position Mechanical Rotary Joystick Interface



Design and development by Jason Penney
Additional revisions based on ideas contributed by Joel Simpson

Features

- Converts the 13 pin harness of 12-position mechanical rotary joysticks into Clockwise and Counter Clockwise output (“dial” or “twist”) signals for a keyboard encoder
- Allows 1 or 2 rotary joysticks to be connected and monitored independently
- Provides more authentic game play than using an *optical* rotary joystick for games designed around the 12-position rotary joystick
- Can simulate an optical rotary joystick’s freewheeling spinner action when the emulator sensitivity and speed settings are adjusted for extra response
- Output switches operate exactly like normally open pushbutton switches (such as standard arcade buttons)
- Straightforward operation: Plug 1 or 2 joystick cables into the interface, wire the outputs to keyboard encoder inputs, and connect a 5 volt power supply:



Description

The objective of this product is to adapt the 12 position mechanical rotary joysticks, translating physical rotations into normal Clockwise and Counter Clockwise outputs compatible with keyboard encoder inputs. Each time the joystick shaft is rotated, the appropriate player’s clockwise or counter clockwise output will activate a keyboard encoder’s input. The keyboard encoder detects the button press and sends the proper keystrokes to the computer, to activate the Dial and Twist keys (the emulator must be configured to map the dial/twist keys to the keyboard inputs used for the rotary interface).

Uses

The rotary interface (RI) may be used to allow the special **12 position mechanical** rotary joysticks (not the *optical* rotary joysticks, that’s a different type of connection) to work with any game that supports buttons for character motion in the clockwise and counter clockwise directions. Once the device is connected, all adjustments are made from within the emulator and the unit becomes transparent. The sensitivity and speed of the rotation commands may be individually tweaked in the emulator to set each game for optimal performance with this interface.

How The RI Acts Like An Arcade Button

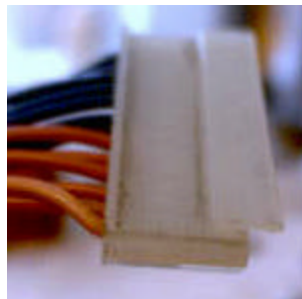
In a physical arcade switch, there is normally no connection between the two switch contacts. When you press the switch, the circuit path between the two contacts is closed. When an arcade switch is wired to a keyboard encoder, one contact on the switch is wired to a designated input on the encoder, while the other is wired to ground. From the perspective of a keyboard encoder, all you need to do to activate a keystroke for any given input is take the input and ground wires and short them together manually instead of using an arcade switch. The rotary interface does this electronically.

The Joystick Cable

The wires on the joystick cable may exist in different color schemes. One known scheme contains a rainbow of colors, while another contains mostly blue and red wires, with a black and orange wire on either end. Both types have the same “R” marking on one end of the cable. **This is the end that must plug into the joystick.**



The other end plugs into the RI board. The wires are connected in a different order on each end of the cable, so the correct orientation is essential for operation. The cable is inserted with the hooded side of the plug facing up, and the exposed metal contacts facing down on both ends of the cable (joystick side and interface board side). Connecting the harness backwards will not harm the interface, but it won't work until it's correctly installed.



The cables are plugged in with the hood facing up

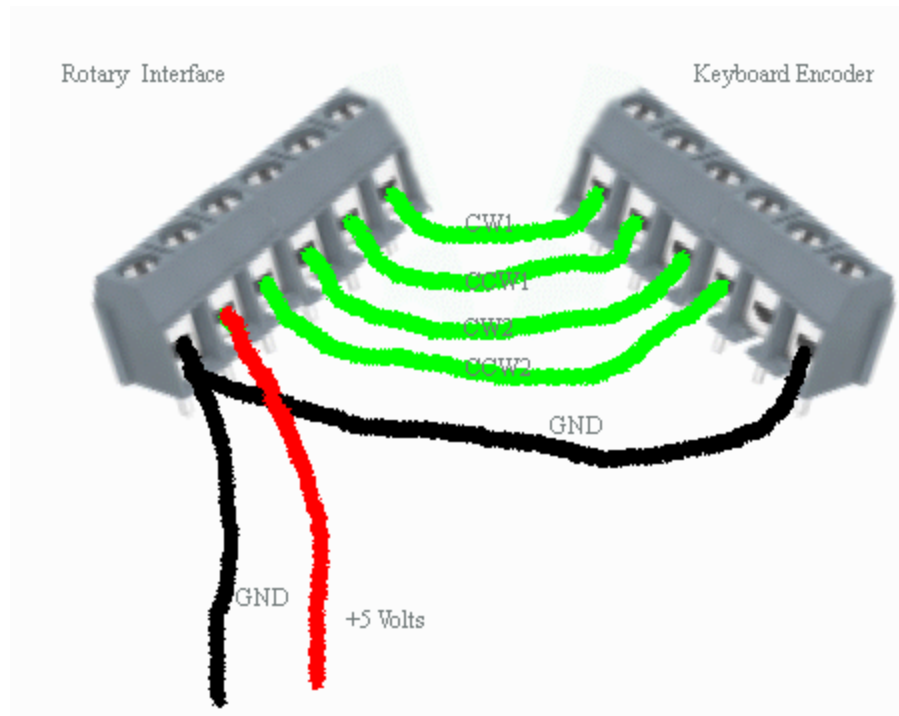
Setting Up The Interface

To plug the joystick cable into the interface, line up the pins as best as possible and push the connector in. It may be easier to insert the connector with a side-to-side motion, as long as the pins aren't subjected to too much stress. The interface uses static sensitive ICs. Avoid directly touching any pins or contacts on the interface, or

laying it on surfaces that may be a static hazard (carpet, clothing, etc). The board should be handled by the edges and treated like any other static sensitive computer card.

Connecting To A Keyboard Encoder

Shown below is a simple connection diagram to illustrate how to wire the RI to a keyboard encoder operating in direct mode. The labeling on the RI's screw terminals may not match this illustration. The boards are properly labeled.



Rotary Interface connected to a keyboard encoder

In this example, both players have their outputs wired to a keyboard input, and the ground wire is linked between the RI and the keyboard encoder to complete the circuit.

Power Supply

The circuit is powered from a +5 volt supply, which may be tapped from another part of the computer power supply, or keyboard power source. Ideally the keyboard encoder circuit would be the best place to tap power, automatically sharing the same system ground connection for the switches. If a separate power source is used, connect a ground wire between the RI and keyboard encoder.

Another source of power may be obtained from a computer power cable splitter used for hard drives. Cut the connector to obtain exposed wires for +5v and ground. ***It is important to connect the power supply to the proper terminals and in the proper polarity, with ground-to-ground, +5 to +5 to avoid damage to the interface.*** It is recommended to verify the polarity and voltage level with a meter before connecting the power source to the interface.

Initial Functional Test

When at least one joystick has been connected to the interface, and it has been given a power source and wired to a keyboard encoder, the unit may be tested in a game, or simply by loading a text editor and rotating the joystick(s) to determine if the keystrokes are being registered. If there is a problem, troubleshooting of the

interface board may be required to ensure that the interface is installed correctly and that it is functioning as intended (see appendix).

Configuring For Optimal Performance

Some of the games respond differently to the rotary dial/twist buttons than others, and different computer configurations/speeds may also require tweaking to find the best response settings. Some games will seem to respond slowly to rotations, while other games may over-trigger and seem to rotate too many times for each single joystick rotation. To set the proper response characteristics for each game, the configuration menu will have to be accessed for each game, and trial and error will be required to get the character responding properly.

By accessing the configuration menu with the “Tab” key, then the “Analog Controls” menu option, the Dial or Twist buttons for player 1 and player 2 may be fine-tuned individually for each game.

If the character begins to auto-rotate when you just advance with one joystick rotation, the sensitivity or speed is set too high, so you must change the settings to make the game react slower to the inputs. If the case were such that the joystick rotations weren’t being picked up every time reliably, you would adjust the dials to make them more sensitive, allowing the input signals to be detected on every hit.

The best way to calibrate the buttons is to start a game, move the character somewhere other than the center of the screen (so it is not hidden by the configuration menu that pops up), then enter the dial adjustment menu in the Analog Controls section of the Tab menu. Use the arrow keys to change the dial settings, and then rotate the joystick to see the character’s responsiveness to each change until it is optimal. The new settings should automatically be saved for each game when you exit.

Once all games have been calibrated uniquely, they may all be played with optimal results every time without further tweaking necessary.

It is important to note that even when the games are tweaked to work optimally, there may be instances where the game character doesn’t seem to respond properly. These are physical limitations of the emulator software and/or the joystick itself.

One such possibility occurs when the joystick is rotated one position, but the character does not respond, or the character may over-trigger and rotate more than once, while otherwise normally responding with one movement per rotation. This is an issue with the emulator itself, as would be noticed if the dial keys were pressed on the keyboard instead of with the rotating joystick. Keeping in mind that this is an emulator to begin with and is prone to imperfect reproduction, if you press a dial key on the keyboard repeatedly, at a fixed rate, eventually you may find that one of your key presses went undetected, or the character double-rotated in that direction. The rotary joystick may thus provide the same result.

If the joystick is rotated at accelerated speeds (beyond what is normally required for gameplay), you may notice the character rotating in both directions even though you are only rotating in one direction on the joystick. This is just switch bounce causing false triggering between locked positions on the joystick. It isn’t an issue for normal joystick usage but in case you observe it, that is the reason for it. For normal rotating speeds, the switches are adequately debounced.

A Note About The Output Switch Closure Timing

The output switch terminals of the RI are configured to be closed (pressed) for a very short duration, and then released. There is no time delay imposed after releasing the switch. What this means is each time the joystick is rotated, if the rotations are occurring fast enough, the on/off cycles will be so fast that it may seem as if the output switch on the RI is being held in the pressed position for that whole rotating duration, instead of being

rapidly opened and closed. This has been found to enhance performance in the emulator in some instances where a quick response is required to rotate a character a great distance. The activity will act like an optical rotary joystick, or a spinner, with the character spinning as fast as the dial keys have been configured to rotate. This gives the player complete control over the character's motion based on how fast the joystick moves. The slower the joystick is rotated, the more control there is over what happens.